Overview of FIR filtering, Windowing, DT systems and DT approximation of CT Systems

Statement of Autonomy: *I affirm that I have composed the term paper autonomously and have used no sources and aids other than those indicated.* **Kempten, 11-Jan-2021**

Signature, Place, Date

DFT-Based FIR Filtering [Linear & Circular Convolution, Overlap-Add]

Linear Convolution via Circular Convolution: Filtering in time domain is implemented by linear convolution of input signal $x(n)$ and the impulse response of the filter $h(n)$, while in frequency domain it is translated as multiplication of both sequences. We are interested in frequency domain because of computational feasibility using digital computer and lower effort[1]. However, it is not feasible to get DTFT because it requires infinite number of samples which is beyond the capabilities of digital computer, therefore DFT is done via FFT algorithm, and DFT gives the imperfect view of the true DTFT of the truncated signal. Furthermore, multiplying the DFT of both sequences and then taking its IDFT doesn't give us the ordinary linear convolution, instead we get the circular convolution[2]. Also, it is wrong to assume that linear time-invariant filtering can be performed by multiplying computed DFT of both sequences because DFT is not the frequency response of either input signal or the filter rather it is the sampling of the frequency response in the θ domain. The circular convolution of two length-N sequence is itself a length-N sequence. It can be imagined as thought the sequences are defined on points on a circle, rather than on a line[3]. Therefore, the circular convolution is also called cyclic convolution or periodic convolution. In order to get the linear convolution instead of circular convolution, N-point FFT is computed by zero padding both the sequences to the length of linear convolution ($N \geq L+M-1$). As the N-point DFT of the output sequence is enough to represent it in the frequency domain, the multiplication of the N-point DFTs followed by the N-point IDFT gives us the same result as we would have obtained by linear convolution.

Filtering of Long Data Sequences (Overlap-Add): Practical real time signal processing applications demand filtering of very long input signals $x(n)$ which leads to long delays because of wait for whole signal availability and limited computer memory. To resolve this, with fixed length (M) of $h(n)$, the $x(n)$ is divided in blocks with length ($L > M$) and then N-point DFT and IDFT is performed over each block which are finally overlapped and added.

Practical Spectral Analysis and Windowing [Distortions, Frequency measurement issues]

Rectangular Windowing and spectral leakage: Spectral analysis of a given discrete-time signal is performed to analyze the presence of sinusoids and their frequency. Ideal FFT is obtained when integer number of periods of a periodic signal are measured but when non-integer periods are measured spectral leakage is added to the FFT. To analyze an analog signal, it is first passed through an antialiasing filter and then sampled at rate $F_s \geq 2B$. Since, practically full length DFT is not feasible because of speed and storage limitations, therefore the duration of the signal is limited to the time interval $T_0 = NT$, where NL is the number of samples and T is the sample interval. Limiting the signal $x[n]$ to N samples in interval ($0 \leq n \leq N-1$) is equal as multiplying $x[n]$ by a rectangular window $w_r[n]$ of length N[3]. Which means by applying FFT without windowing, rectangular window is automatically enforced and this window in frequency domain is sinc in nature. As a result, in addition to the lobe due to the frequency components, additional lobes are also appeared in the DFT spectrum. This rectangular windowing is implemented by convolution of FFT of original signal and Dirichlet Kernel (function in FFT of rectangular window). Since FFT of this rectangular window is not ideal (an impulse function) we get distortions (spectral leakage) i-e *smearing* (blend of periodic components) and side-lobe *interferences* (strong component's side lobe masked over weak component's main lobe).

Different window types to minimize distortions: So now we can define *window* as a function that is non-zero in a finite range of time[4] and *windowing* truncates a signal to that finite length and reshapes it[2]. Precisely, the window function modifies the amplitude and frequency response of the sampled data[5]. The rectangular windowing is mare truncation without any reshaping. To minimize the above-mentioned distortions, we would like the Dirichlet Kernel to have as narrower main lobe and as lower side lobes as possible. The closer the shape of the kernel function to the delta function the lower will be the distortions, but it requires trade-off among the width of the main lobe and level of the side lobes, and both are inversely related. Two main parameters to *evaluate our window* are; (i) width of the main lobe with reference to the width of rectangular window of same length (its $4\pi/N$ for rectangular window and good), (ii) level of largest side lobe with reference to the amplitude of main lobe (it is -13.5.dB for rectangular window and undesirable). *Bartlett window* or triangular window involves squaring of rectangular window kernel function reduce side-lobe level by twice (-27dB, better than rectangular) but doubling the main lobe width to $8\pi/N$ (trade-off). *Hann, Hanning or Cosine window* lowers the side-lobe level (-32 dB, but increase main lobe width to $8\pi/N$, better than Barlett) by superposition and its kernel is achieved by adding three frequency shifted Dirichlet kernels partially cancelling their side lobes. By changing the magnitudes of three Dirichlet kernels *Hamming window* is achieved (-43db, $8\pi/N$, better than Hann). Also known as raised-cosine window, unlike Hann window it has a special property that its highest side lobe is not closest to the main lobe. By further hit and trail over the kernel's magnitudes, *Blackman window* is obtained that further decreases side-lobe level -57db (better than Hamming) but have main lobe width $12\pi/N$ (trade-off, thrice of rect). Finally, *Dolph* and *Kaiser* are the modern windows and Kaiser is the most popular. The main lobe width and side-lobe level is controlled by factor α in the Bessel function of the Kaiser window. For α=3,6,9,12, main lobe width is 6,8,12,16π/N (increase in trade-off while α increase) and side lobe level is -25,-45,-65,-90dB (decreasing as α increase), respectively. By windowing, our desire is to decrease the width of main lobe and increase the flatness of side-lobe level and either one is obtained over the cost of other, so we need to compromise and it depends on the type of *application*, e.g[5] Blackman is used for ADC characterization, Hann is used for vibration data (narrowband random signal), sine wave or combination of waves, Hamming is used for closely spaced sine waves and Kaiser is used for two tones with very different amplitudes and-close frequencies.

Frequency Measurement Problem: FFT deconstructs a time domain signal to frequency domain components in discrete values (also known as bins). Frequency domain gives us voltage present at different frequencies present in original signal. Achieving these accurate frequencies are very significant for many *applications*[6] such as by deconstructing the radio waves we can listen to the particular station, earthquake vibrations we can optimize building designs to withstand strong vibrations, PC data and skipping least significant frequencies to save memory (file compression), bio-optical signals[7] to improve diagnosis, doppler shifted signal from radar/sonar. When two sinusoid components are closely located i-e the frequency difference is small, the peaks repel each other which deviates them from their true locations, this effect is known as *biasing* and if the frequency difference keeps decreasing the peaks start to merge with each other and eventually the weaker component becomes invisible. Side lobe interference is also another problem because the side lobes of the Dirichlet kernel are relatively high and if amplitude of one frequency component is very small it becomes impossible to detect it. We can see that depending on distance between frequencies and side lobe levels, we encounter different issues and that can be addressed with different windows. Few possibilities are summarized in Table 1.

Table 1. Different windows to address different frequency measurement problems

Frequencies location	**Amplitude of components**	**Problems**	**Solution (Window)**	**Compromise, Threats**
Close	~ same	Smearing	Rectangular: to lower ML width	Increase in lobe level: can lead to interference
Close but not<MLWidth	One is weak	Interference	Kaiser: to lower SL level	Increase in ML width: lead to smearing, compensated by increasing N (no. of samples) but increase computation
Close < ML Width	One is weak	smearing, Interference	Hamming: to best fit among ML and SL level	Compromise between ML width and SL level
Not so close, >ML Width	~ same	Interference	Kaiser: to lower SL level	Increase in ML width: can lead to smearing
Not so close, >ML Width	One is weak	Interference	Kaiser: to lower SL level	Increase in ML width: can lead to smearing
Close	Same/different	Biasing, interference	Kaiser: to lower SL level	Increase in ML width: can lead to smearing
Unknown	Unknown	Mostly: interference	Choose Kaiser and vary values of α	Compromise between ML width and SL level

Also, there are two *instructions to choose a window* provided the prior information about signal is available, first; "keeping the expected frequency difference greater than main lobe width" but it can be only be done if we have prior information about that signal. Therefore, it is not possible to differentiate two frequency components separated by frequency less than $1/T_0$ and this is the reason for aliasing effect and Nyquist rule (sample rate should be twice the bandwidth) is applied to avoid aliasing error. The second instruction is "keeping SL level (dB) higher than maximum amplitude separation of sinusoidal components". So, we can see that there is no absolute way to choose a window and based on the nature of components we have to choose the window that best describes our signal.

DFT of Noisy Signals: Due to noise in signal, the accuracy also depends on signal to noise ratio. The accuracy can be improved by using special DFT forms such as zero-padded DFT or chirp Fourier transform.

DT Systems [DT filters, Linear Phase FIR Filters]

DT-Filters: A filter allows the signals to pass in certain frequency range and attenuates in other frequency bands. Filtering is significant in DSP e.g for spectral shaping and analysis, detection of radar/sonar signals, and noise removal and numerous applications in power supplies, audio electronic and radio communications. Based on the availability of feedback, filters can be (Table 2);

Table 2. IIR, FIR Filters, their advantages and disadvantages

Recursive	Closed loop i-e have feedback, past input (M) and output (N)	are IIR (infinite impulse response filters)	stability issues exist as poles not inside the unit circle	work faster, require less memory space, suitable for amplitude monitoring	Issue: Nonlinear phase response
Non-recursive	Open loop i-e no feedback, only past input values (M)	are FIR (finite impulse response filters)	always stable, poles at origin of Z-plane	good for linear phase applications, stable designs, flexible amplitude response	Large delays, complex computation ~4 x > IIR

A filter has a Passband (allowed range of frequencies), and *Stopband* (block/stop range of frequencies), and *Transition-band or skirt* (range of frequencies allowing transition between pass-band frequency (Ωp) and stop-band frequency (Ωs). Practically in filter's passband there are amplitude vibrations called as passband ripples (δ_p), and there is stopband attenuation (δ_s) is the minimum attenuation level in the stop band of the filter. In bandpass filter and bandstop filter there exist attenuation (δ_{s1}) and attenuation (δ_{s2}) on right and left side of the passband and stopband, respectively. For all different filter types (Table 3) by reducing these ripples, attenuations and frequency gap, these can be made ideal.

Table 3. Types of filters

Type	Filter Function	To make ideal
Low Pass	allows frequencies lower than certain limit to pass through	δ_p, δ_p → (tends to) 0, Ωs → Ωp
High Pass	allows frequencies higher than certain limit to pass through	
Band Pass	allows frequencies between certain lower and higher limits to pass through	δ_p, δ_{s1}, δ_{s2} → 0, Ωs → Ωp
Band Stop	attenuates frequencies outside certain lower and higher frequency limits	

Linear Phase FIR Filters: Most FIR filters have linear phase i-e the phase response of the filter is a linear (straight-line) function of frequency that allow all frequency components of an input signal to pass through filter with the same delay. These filters preserve the time domain characteristics (shape) of input signal and have potential applications e.g heart monitor (crucial to preserve filter shape), IoT devices (to maintain time domain representation of signal), avionics. They are categorized into following four main types Table 4;

Table 4. Types of Linear Phase FIR Filters

Filter Type	Filter Order	Filter Length M+1	Impulse resp. coefficients	Usage
Type-I	even	Odd	Symmetric	LP, HP, BP, BS and Multiband
Type-II	odd	Even		LP, BP
Type-III	even	Odd	Asymmetric	Differentiators, Hilbert transformers
Type-IV	odd	even		

Different methods for designing FIR filters are summarized in Table 5.

Table 5. Methods for designing FIR filters

Method	Implementation	Advantages	Issues, constraints	Solution(s)
Windowed sinc	windowing by impulse response (attenuating Gibbs oscillations) **Matlab:** h=fir1*	Simple, good filter performance and robust.	Ripples, ⇒ Ripple decay ∝ filter order	Chebyshev/Kaiser win. to increase decay (stopband attenuation) but it widens transition band
			wide transition band	Increasing filter order (M) but it increases computation and delays
Frequency sampling	Impulse response→DFT→… …→IDFT→window to smooth the impulse response. **Matlab:** b = fir2**	Can be implemented using lower filter orders, flexible designs, moderate computation	specifications not guaranteed, huge deviations between the sampling points	hit and trial to until specs are met, increasing the sampling rate (dense the grid) but it will increase computation
Parks-McClellan design[1]	Lowering the maximum ripple in all bands of interest **MatL[1]:** = firpmord & =firpm***	smaller filter length, remove noise with least possible signal distortion	filter design might fail to converge	Increasing filter order, Soothing the design by widening transition band and lowering attenuation[8]

*h=fir1(M,omega_c,window(M+1,options), **function b = fir2_min(nn, ff, aa)
***[N,fo,ao,w]=firpmord(f_spec,AA,dev,Fs) for order and other parameters; b=firpm(N,fo,ao,w) for filter weights (design coefficients)[1] (DSP-Lab 6)

DT Approximation of CT Systems [DT IIR from CT filters, implement in Matlab]

The design of filters is done is 3 steps[9]; 1. Specifying required properties of the system, 2. Approximating the specifications by DT system, 3. Implementation of system. The 1st and 3rd step depend on application and implementation technology, respectively. The 2nd step is primary focus for filter design. Practically, filters are implemented via digital computers to filter signals obtained from CT signals by analogue to digital conversion (ADC) and then periodic sampling.

Design of DT IIR Filters from CT Filters: Discrete time infinite impulse response (DT IIR) filters are implemented by transforming the CT filter to DT filter following certain conditions. This approach is preferred because CT IIR design procedures have simple closed form CT design formulas and its useful to utilize techniques already matured for CT filter, also the advance CT IIR filter design methods can produce promising results. But, if these procedures are applied directly for DT IIR filters, these simple closed form design formulas are not achieved. Therefore, the 1st condition in this transformation is; the basic properties of CT frequency response must be kept preserved in the resulting DT filter, which means to map the imaginary axis of s-plane over the unit circle of *z*-plane. The 2nd condition requires the stable CT filter to transform into stable DT filter, which means that the DT filter must have the poles inside the unit circle in the *z*-plane if the CT system has the poles in the left half of the *s*-plane.

Impulse Invariance Method: Impulse invariance refers to defining DT system by sampling impulse response of CT system and impulse invariance method can be defined as approximating the CT system's impulse response by DT convolution (involves summation) instead of CT convolution (that involves integral). To accurately compute the integral, while sampling the CT impulse response quite small samples are required. Except aliasing, in this method, the shape of the frequency response is preserved as there exist a linear relation between CT and DT frequency. Therefore, impulse invariance method is appropriate only for bandlimited filters because severe aliasing distortion is observed in highpass CT filters and additional bandlimiting is required to avoid that. Implementing this DT system as a FIR may yield a very long FIR filter resulting in higher computation efforts.

Tustin's Method (Bilinear Method): In this method instead of impulse response, the frequency response of the DT system is approximated to the frequency response of the CT system. Furthermore, to avoid the aliasing issue discussed above, bilinear transformation is applied. It involves algebraic transformation among variables s and z that map s-plane's entire imagineray axis to one revolution of the z plane's unit circle. This transformation must be non-linear as $-\infty \leq \Omega \leq \infty$ is mapped onto $-\pi \leq \omega \leq \pi$. Aliasing is avoided by taking the whole frequency range $(-\infty \ldots\ldots \infty)$ to $-\frac{F_s}{2}$ *to* $\frac{F_s}{2}$. An example for Tustin's application is single tone tuning of a guitar's amplifier. A variac is used in its analogue circuit to control the resistance and output. This can be controlled by removing high frequency content using the filter that muffles the guitar's sound.

Butterworth, Chebyshev-1, 2 and Elliptic filters:

These are the typical continuous time approximations and have different characteristics in passband and stop band (Figure 1) and can be easily implemented in Matlab (Table 6).

The Butterworth and Cheby2 have flat passbands and wide transition bands while Cheby1 and Ellip roll off faster but exhibit passband ripples.

Table 6. Characteristics of filters in passband and stopband

Filter	Passband	Stopband	Matlab Syntax
Butterworth	monotonic	monotonic	`[b,a] = butter(n,Wn)`
Chebyshev 1	equiripple	monotonic	`[b,a] = cheby1(n,Rp,Wp)`
Chebyshev 2	monotonic	equiripple	`[b,a] = cheby2(n,Rs,Ws)`
Elliptic	equiripple	equiripple	`[b,a] = ellip(n,Rp,Rs,Wp)`

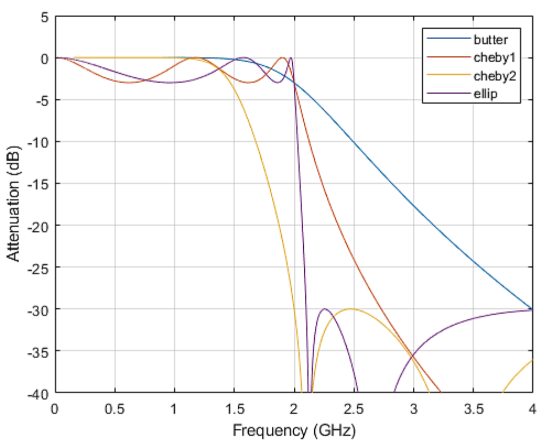

Figure 1. Characteristics of butter, cheby, elliptic[8]

References

1. Brückl, S., 2020. DSP Lecture Notes and Lab Manuals, M-EE UAS Kempten.
2. Porat, B., 1997. A Course in Digital Signal Processing. New York: John Wiley.
3. Proakis JG, Monolakis DG. Digital Signal Processing: Principles, Algorithms, and Applications.; 1996.
4. David M. Harrison. Sampling & Windowing. 1999:1-7. http://spider.ipaccalte-ch.edu/staff/fmasci/home/astro_refs/SamplingAndWindowing_Fourier.pdf.
5. Chioye L., 2018. FFTs and Windowing by Texas Instruments (IT Prec. Labs).
6. Understanding FFTs and Windowing by National Instruments. 2019.
7. Romo-Cárdenas G, et al. Nyquist-Shannon theorem application for Savitzky-Golay smoothing window size parameter determination in bio-optical signals. Results Phys. 2018. doi:10.1016/j.rinp.2018.08.033.
8. Mathworks.com. Matlab and Simulink. [online] [Accessed 2 January 2021].
9. Oppenheim A V, Schafer RW. Discrete Time Signal Processing 2nd Edition. Book. 1998.